Package: rtemis.server (via r-universe)

May 27, 2026

Version 0.0.6

Title 'rtemis' Server

Date 2026-05-26

Description Create a websocket server to run 'rtemis' functions from an 'rtemislive' client.

License GPL (>= 3)

URL https://www.rtemis.org, https://live.rtemis.org

BugReports https://github.com/rtemis-org/rtemis.server/issues

ByteCompile yes

Depends R (>= 4.1.0)

Imports methods, utils, S7, data.table, rtemis (>= 1.1.0), rtemis.core, jsonlite, later, mirai, nanonext, openssl

Suggests arrow, callr, testthat (>= 3.0.0), uuid

Encoding UTF-8

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

LazyData true

Config/roxygen2/version 8.0.0

Config/pak/sysreqs cmake libssl-dev

Repository https://rtemis-org.r-universe.dev

Date/Publication 2026-05-27 15:33:55 UTC

RemoteUrl https://github.com/rtemis-org/rtemis.server

RemoteRef HEAD

RemoteSha c3bdae945259cf7891a546bd4379e25a8b3550a1

Contents

`rtemis.server-package` *rtemis.server: rtemis Server*

Description

Local WebSocket server that bridges browser-based rtemislive clients to a persistent R session running rtemis. See `vignette("rtemislive")` (TODO) and `specs/` for the wire protocol.

Details

rtemis.server: rtemis WebSocket server

Author(s)

Maintainer: E.D. Gennatas <gennatas@gmail.com> (ORCID) [copyright holder]

Authors:

- E.D. Gennatas <gennatas@gmail.com> (ORCID) [copyright holder]

See Also

Useful links:

- https://www.rtemis.org
- https://live.rtemis.org
- Report bugs at https://github.com/rtemis-org/rtemis.server/issues

forward_progress *Forward a* rtemis::train *progress checkpoint over the msg sink*

Description

Thin adapter passed as progress = to rtemis::train(). Calls rtemis's internal msg() with caller = stage, so the daemon-side sink (installed by init_daemon_progress) ships an envelope whose caller field carries the structured stage name (e.g. "outer_fold"). The host turns that into a job.progress event with data$stage set, which the UI can route on without text-matching the message.

Usage

```
forward_progress(stage, current, total, message)
```

Arguments

stage	Character scalar: Structured stage name. Becomes the caller field on the wire envelope (e.g. "outer_fold").
current	Integer: 1-based index of the checkpoint. Unused by this adapter directly (encoded into message upstream), kept in the signature so it matches the rtemis::train progress contract.
total	Integer: Total checkpoints. Same as current - present to match the contract.
message	Character scalar: Human-readable line, e.g. "Outer fold 2/5". Becomes the envelope's text field.

Details

msg is unexported from rtemis; the reference is bound at package source-eval time in 00_init.R via getFromNamespace, so calling msg() here avoids both rtemis:::msg (R CMD check NOTE) and any per-call namespace lookup.

Designed to be referenced as rtemis.server::forward_progress inside the mirai job expression - mirai loads rtemis.server on the daemon on first use, sourcing 00_init.R once, so the msg binding exists before the callback is ever invoked.

Value

Invisible NULL.

Author(s)

EDG

`serve` *Start the rtemislive backend*

Description

Launches a local-only WebSocket server that bridges the rtemislive browser frontend to a persistent R session running rtemis. Provides async training, real-time progress, structured result transfer, and session-aware state across reconnects.

Usage

```
serve(
  port = NULL,
  host = "127.0.0.1",
  n_daemons = 1L,
  origins = NULL,
  token = NULL,
  heartbeat_interval = 5,
  session_ttl = 86400,
  data_ttl = 3600,
  gc_interval = 60,
  tick_ms = 50L,
  max_concurrent = n_daemons,
  max_sessions = 16L,
  verbosity = 1L
)
```

Arguments

`port`	Integer: TCP port to listen on. Must be in `1024:49151`. Defaults to `5757`, or the value of `RTEMISLIVE_PORT` if set.
`host`	Character: Bind address. Defaults to `"127.0.0.1"`. Server refuses to start on any other address.
`n_daemons`	Integer: Number of mirai worker processes. Default `1L`: rtemis training jobs already parallelise internally (OpenMP for gradient boosters, parallel tuning, parallel outer resampling), so a single daemon runs one job at a time with full core access. Increase only when you want multiple simultaneous jobs and accept that each gets a fraction of the cores.
`origins`	Optional character vector: Allowed `Origin` headers on the WS upgrade. `NULL` uses the spec defaults (`live.rtemis.org`, `localhost:3000`, etc.).
`token`	Optional character scalar: Auth token clients must present. `NULL` generates a fresh 8-byte random token.
`heartbeat_interval`	Numeric, seconds: Heartbeat tick rate.
`session_ttl`	Numeric, seconds: Idle-session GC TTL.

`data_ttl`	Numeric, seconds: Idle-data-handle GC TTL.
`gc_interval`	Numeric, seconds: How often GC runs.
`tick_ms`	Integer milliseconds: Background tick rate for the non-WS periodic work scheduled via `later`. Default `50`.
`max_concurrent`	Integer: Cap on concurrent jobs across all sessions. Defaults to `n_daemons`: no point queuing more running jobs than there are workers.
`max_sessions`	Integer: Cap on the number of sessions.
`verbosity`	Integer: `>= 1L` prints the startup banner.

Details

Blocks until the user interrupts (Ctrl-C) or another mechanism sets `server$stop_requested`. See `specs/rtemislive.md` for the wire protocol and architectural details.

Value

Server env, invisibly. Returned after the loop exits so callers (notably tests running the server on a mirai task) can inspect state.

Author(s)

EDG

See Also

`shutdown()`

Examples

```
## Not run:
# Run on the default port; Ctrl-C to stop.
serve()

# Multiple workers for a multi-user setup (each job gets fewer cores).
serve(
  port = 5757L,
  n_daemons = 4L,
  origins = c("http://localhost:3000"),
  token = "abcd-1234-ef56-7890"
)

## End(Not run)
```

shutdown *Signal a running rtemislive server to stop*

Description

Sets `server$stop_requested`. On its next tick the loop will close the HTTP listener (which causes `http_server$serve()` to return) and `serve()` cleans up daemons and sockets via `on.exit`.

Usage

```
shutdown(server)
```

Arguments

`server`	Server env returned (or shared) from `serve()`.

Details

Useful when the server runs on a mirai task or a separate R process that can be passed the server env (e.g. in integration tests). For a server running in the user's foreground R session, just press Ctrl-C.

Value

`NULL`, invisibly.

Author(s)

EDG

Examples

```
## Not run:
# Start the server on a mirai task so it doesn't block the session.
task <- mirai::mirai({
  rtemis.server::serve(port = 5757L, verbosity = 0L)
})

# ... do work ...

# Signal the server to stop gracefully.
rtemis.server::shutdown(task$data)

## End(Not run)
```

Index